

Métricas para el Diseño Orientado a Aspectos

L. Baigorria, C. Salgado, M. Peralta, D. Riesco, G. Montejano
Departamento de Informática
Universidad Nacional de San Luis
San Luis, Capital, Argentina
Ejército de los Andes 950 – San Luis – San Luis – Argentina
C.P.: 5700
Tel.: 54-02652-424027 – Int. 251

[flbaigor, csalgado, mperalta, driesco, gmonte]@unsl.edu.ar – web: <http://www.sel.unsl.edu.ar>

Resumen

El desarrollo de grandes sistemas de software es una actividad abordada por la Ingeniería de Software desde diferentes puntos de vista y desde sus inicios. Cada día surgen nuevas técnicas y metodologías que intentan mejorar la calidad y la eficiencia del software. La orientación a aspectos (OA) es una nueva técnica aplicada en el desarrollo de software la cual busca mejorar la calidad del mismo.

En la actualidad se aplica la OA en grandes proyectos es por eso que la evaluación, de su uso y las eventuales mejoras, es crucial.

Las métricas de software son ampliamente utilizadas para evaluar objetivamente el software. Existen gran cantidad de métricas definidas para la orientación a objetos (OO) y otros paradigmas pero muy pocas para la OA. Es por esto que se realizó la definición métricas para el desarrollo OA, en particular para la etapa del diseño, permitiendo al ingeniero de software tener una visión objetiva del diseño realizado.

Palabras Clave: Orientación a Aspectos, Métricas, Diseño Orientado a Aspectos, OA.

Contexto

El presente trabajo se enmarca en el Proyecto de Investigación: Ingeniería de Software, Conceptos, Métodos y

Herramientas en un Contexto de “Ingeniería de Software en Evolución”, Proyecto de Ciencia y Técnica 22F822, periodo 2008-2010 – Facultad de Ciencias Físico-Matemáticas y Naturales, Universidad Nacional de San Luis.

Introducción

El objetivo de la Ingeniería de Software es construir un producto de software o mejorar alguno existente. Un proceso de desarrollo de software efectivo proporciona normas para la construcción eficiente de software de calidad. Este proceso está compuesto, a grandes rasgos, por cinco etapas: Requisitos, Análisis, Diseño, Implementación y Prueba. Uno de los objetivos de la etapa de diseño es crear una abstracción de la implementación, en el sentido de que la implementación como un refinamiento directo del diseño. Esto permite utilizar tecnologías como la generación de código y la ingeniería de ida y vuelta entre el diseño y la implementación, entre otros. Es por esto que el diseño está directamente relacionado con los lenguajes de programación y es en ésta etapa, donde se construyen modelos que dependen de ellos.

Por otro lado, uno de los principales problemas en el proceso de desarrollo de software es el uso de grandes cantidades de recursos. Aún con el

incremento de la automatización de diferentes actividades del desarrollo de software, los recursos son escasos y costosos. Como consecuencia de esto y en la constante búsqueda de mejoras en el desarrollo de sistemas de software, los investigadores y desarrolladores están utilizando la orientación a aspectos (OA) en el proceso de desarrollo [1].

Si bien la OO es uno de los avances más importantes en la ingeniería de software para la construcción de sistemas complejos, utilizando el principio de descomposición y reutilización. Muchas veces las descomposiciones causan ejecuciones ineficientes. Éstas surgen debido a que las unidades de descomposición no siempre van acompañadas de un buen tratamiento de aspectos como: sincronización, manejo de errores y de excepciones, administración de memoria y gestión de seguridad. Estos aspectos que entrecruzan y atraviesan todo el sistema, conocidos como crosscutting concerns, no pueden ser totalmente separados con las técnicas de OO.

Surge entonces la OA, la cual tiene como unidad principal el aspecto. Los aspectos encapsulan los crosscutting concerns que afectan a distintos componentes del sistema, es decir se encuentran en varias partes de un programa de manera tal que se reduzca la dependencia, y se aumente su reutilización.

Como se dijo anteriormente la industria del software ha atravesado una serie de etapas a lo largo del tiempo desde sus inicios hasta llegar al desarrollo rápido de calidad requerido en los noventa. La calidad de software es un punto crucial en cualquier desarrollo serio.

Para determinar si un software es de calidad o no se pueden utilizar métricas de software las cuales son también

aplicables a otras áreas como prueba, administración y mantenimiento del software.

En los últimos años existe un gran interés en las métricas de software debido a su potencial para predecir el uso más eficiente de los recursos y mejorar la calidad. Se han realizado una gran cantidad de trabajos relacionados a las métricas, en especial aplicadas a la OO y procedimentales [2,3], pero hay muy pocos trabajos realizados con métricas específicamente para sistemas OA. Algunos investigadores han establecido que si se aplica la metodología de OA para el desarrollo se mejoran los valores obtenidos en las métricas para OO [4].

Se sabe que las métricas de software son formas de calificar el diseño de software. Se puede decir entonces que las métricas aplicadas a la OA son cruciales para determinar la efectividad de su uso.

Tanto el diseño como el modelado de sistemas ocupan un lugar importante en la Ingeniería de Software. Los modelos brindan la posibilidad de abstraer sistemas y facilitar la implementación. Por otra parte, la construcción de buenos modelos asegura un correcto desarrollo de la arquitectura del sistema. Es imprescindible entonces contar con herramientas que permitan evaluar la calidad del diseño, es por esto que la definición de métricas específicas a la OA es indispensable.

La definición de métricas OA permitirá evaluar objetivamente el desarrollo de un producto de software de manera tal que se logre aprovechar al máximo las ventajas de la OA.

Considerando lo antes expuesto, se realizó la definición de métricas específicas para sistemas OA, aplicables a partir de la etapa del diseño. En general es en esta etapa donde surge el modelado de los aspectos del sistema.

La mayor parte del diseño OA es subjetivo, un diseñador “sabe” como caracterizar un sistema OA para implementar efectivamente los requerimientos del cliente. Pero una vista más objetiva del diseño sería más beneficiosa. La aplicación de métricas provee esta vista objetiva del diseño realizado [5].

Además, si se tienen en cuenta las similitudes entre OA y OO algunas métricas como las C&K [6] o las métricas MOOD [7], sirven de base para la definición de métricas para sistemas OA.

Uno de los inconvenientes encontrados al momento de definir métricas, para el diseño OA, es que en la actualidad no hay acuerdo con respecto al modelado de sistemas orientados a aspectos. Es por este motivo que se realizó una extensión de UML (Unified Modelling Language) [8] utilizando perfiles UML y OCL (Object Constraint Language) [9], los cuales son estándares de la OMG (Object Management Group) [10] ampliamente conocidos por los ingenieros de software. Debido a esto, la comprensión y utilización del mismo es simple y rápida.

Antes de realizar la definición de las métricas utilizando el perfil propuesto se realizó la aplicación del mismo a sistemas reales para validar su utilidad y efectiva aplicación.

Además se puede situar la definición del perfil en la vista PSM de MDA (Model Driven Architecture) [11] el cual es promovido por la OMG y una temática actualmente muy investigada. En este caso se trata de una vista *PSM de Aspectos* ya que a través del perfil UML, restringido con OCL se pueden realizar modelos OA.

Como se dijo anteriormente el análisis de las métricas aplicadas a distintas etapas del desarrollo de software es un tema que debe ser profundizado, en

particular para la OA ya que su uso en grandes proyectos de software aumenta rápidamente.

Esto ha motivado el análisis de métricas para poder aplicarlas al diseño OA y encontrar además de los posibles efectos, nuevas métricas que ayuden a calificar objetivamente un sistema OA. Se definieron métricas específicas para sistemas OA, aplicables en la etapa del diseño ya que es esta etapa la más costosa y en la cual surge el modelado de los aspectos del sistema.

Para realizar la definición de las métricas se utilizó OCL, el cual es un lenguaje no ambiguo y que permite expresiones lógicas.

Una vez realizada la definición teórica de las métricas se efectuaron pruebas de éstas en modelos OA y realizando de esta manera una validación empírica de las métricas propuestas.

La intención de este trabajo es ayudar a los ingenieros de software que decidan incorporar la OA en el desarrollo de su sistema de software proveyendo, herramientas que permitan evaluarla de manera objetiva.

Líneas de Investigación y Desarrollo

Teniendo en cuenta las ventajas que proporcionan el uso de lenguajes ampliamente conocidas como UML, OCL y la gran cantidad de herramientas CASE que los utilizan, en la actualidad se trabaja para alcanzar los siguientes objetivos:

- Automatizar la incorporación de perfiles a herramientas de modelado UML.
- A partir del uso del perfil OA realizar transformaciones en el marco de MDA para llegar automáticamente al código ejecutable.
- Incorporar perfiles a herramientas que permitan lo permitan y aplicar las métricas definidas de forma

automatizada. Esto es colocando en un archivo las métricas en OCL y aplicarlas sobre el modelo desarrollado.

- Construir una herramienta que a partir de un modelo OA realice el cálculo de las métricas definidas.

- Luego del refactoring de un sistema OO a uno OA realizar mediciones claves como la no existencia de la paradoja entre aspectos o el alto acoplamiento.

Resultados y Objetivos

Se han definido métricas en OCL y se las ha aplicado a diseños OA a través de Papyrus [12], la cual es una herramienta de modelado UML que permite incorporar restricciones OCL. Luego se ha realizado el análisis de los resultados para comparar en forma objetiva diseños OA entre sí y diseños OA vs. OO. De esta manera se provee al ingeniero de software una herramienta automatizada para evaluar el diseño realizado y poder así mejorar la calidad del mismo.

Formación de Recursos Humanos

Siguiendo la presente línea de investigación se han finalizado una tesis de posgrado [13] y una tesis de grado [14]. El objetivo de las mismas es proporcionar herramientas que ayuden al ingeniero de software en el desarrollo OA y permitan evaluar el mismo de manera objetiva.

Referencias

[1] Sommerville I.; “Software Engineering”, 8va. Ed.; Prentice Hall; 2007.
[2] Dines Bjorner: “Software Engineering 2”: Ed. Springer, 2006
[3] Lorenz, Mark and Kidd, Jeff, “Object Oriented Software Metrics”, Prentice Hall Publishing, 1994.

[4] Zhao J.; “Measuring Coupling in Aspect-Oriented Systems”; Department of Computer Science and Engineering; Fukuoka Institute of Technology
[5] Piattini M., Garcia F., Caballero I.; “Calidad de Sistemas Informáticos”; Alfaomega, Ra-Ma; 2007.
[6] Chidamber S., Kemerer C.; IEEE Transaction Software on Software Engineering; pag. 476-493; 1994.
[7] Brito e Abreu F., Melo W.; “Object-oriented Software Engineering: Measuring and controlling the development process”; 4th. International Conference on Software Quality; McLean; USA; 1994.
[8] Object Management Group. <http://www.omg.org>; 2010.
[9] UML 2.3 Superstructure; <http://www.omg.org/spec/UML/2.3/Superstructure/PDF/>; Mayo 2010.
[10] UML 2.0 OCL Specification, <http://www.omg.org>, Agosto del 2009.
[11] Object Management Group. Model Driven Architecture Guide; <http://www.omg.org>; 2010.
[12] Papyrus, sitio oficial <http://www.papyrusuml.org/>, diciembre 2010.
[13] Baigorria L. Tesis de Maestría en Ingeniería de Software.”Definición de Métricas en OCL según el Metamodelo de la OMG aplicadas al Diseño Orientado a Aspectos”; Facultad de Ciencias Físico Matemáticas y Naturales; UNSL. Diciembre 2010. Directores: Riesco D.; Montejano G.
[14] Adbelahad C, “Generación de Código Orientado a Aspectos Partiendo de un Diseño Orientado a Aspectos”; Trabajo de Licenciatura en Ciencias de la Computación; Facultad de Ciencias Físico Matemáticas y Naturales; UNSL; 2010. Directores: Riesco D. Baigorria L.